

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

APPLICATION FOR UNITED STATES PATENT

FOR

RESOLVING MULTIPLE MASTER NODE CONFLICT IN A DDB

Inventors: Sriram Krishnan, Gregory W. Lazar

Attorneys:
Joel Wall, Esquire
P.O. Box 169
Hopkinton, MA 01748
508-435-4432

September 25, 2001

RESOLVING MULTIPLE MASTER NODE CONFLICT IN A DDB

CROSS REFERENCE TO RELATED APPLICATIONS

This patent application relates to another patent application filed on even date herewith having co-inventors Sriram Krishnan, Andreas L. Bauer, and Gregory W. Lazar, assignee in common with this patent application, the title of “Managing a Distributed Directory Database”, and which is incorporated by reference herein in its entirety.

BACKGROUND OF THE INVENTION

1. Field of the Invention:

The present invention relates to apparatus, methodology, systems, and/or computer program product for resolving multiple master conflict in a directory database (DDB) and, more particularly, relates to resolving multiple master conflict in a globally-distributed DDB within a computer network environment such as a client server network.

2. Description of Prior Art:

As may be observed by even the most casual computer system user, computer networks appear to be continuously evolving into ever-more sophisticated communication systems. Accordingly, computer networks are having an ever-increasing impact on modern-day lifestyle for many people, as the Internet is demonstrating. Computer networks include local area networks (LANs) such as might be encountered within a singular and localized corporate business organization where corporate functions

1 (engineering, marketing, sales, advertising, human resources, etc.) are linked together in a
2 localized computer network. Computer networks also include wide area networks
3 (WANs) such as might be encountered within a corporate business organization that is
4 nationwide or even worldwide in scope. The Internet is the premier example of a WAN,
5 but is one that is obviously not constrained within any particular business organization
6 and allows access to virtually any organization or individual. It is widely understood that
7 these networks not only utilize hard-wire bus communication paths to accomplish their
8 objectives, but can also utilize wireless telecommunication links via satellites and the like
9 as well.

10
11 Computer network configurations usable within LANs and WANs include client-
12 server network configurations. A client or workstation (typically having a user interface)
13 can be networked to multiple servers which serve the client in a variety of ways. Such a
14 client-server computer network configuration can be particularly useful in certain
15 applications, such as in managing a computer data storage system. A client-server
16 network can be used with the Internet. Thus, quality of operation of a storage system
17 within a client-server network operatively coupled over the Internet not only can impact
18 quality of performance of that client-server computer network, but can also impact
19 quality of service received by multiple human users connected through the Internet to that
20 client-server network. A failed or degraded storage system thus can escalate into a failed
21 or degraded service for multiple human users.

1 A client-server network today may typically be based on an object oriented
2 computer system which means that such system employs one or more object-oriented
3 computer languages such as C++, XML (eXtensible Markup Language), JAVA, and/or
4 others. Briefly, an object, in computer software terms, also referred to as a “node”, is a
5 dedicated area of memory which can be thought of as an impervious container holding
6 both data and instructions within itself, both defining itself and its relationships to other
7 objects in the computer system or network. Such object or node can send and receive
8 messages to and from other objects, respond and react to such messages (e.g. commands)
9 but shall normally be impervious to internal scrutiny. For example, in the above-noted
10 computer data storage system (a kind of computer) each object (system object) may
11 describe or relate to a specific tangible detail in the storage system or in the storage
12 system’s processor (e.g., details such as those describing or relating to aspects of
13 operation of the processor’s cooling-fan, power switch, cache memory, power supply,
14 disk drive interface, etc.). These tangible objects (nodes) in the storage system can send
15 messages to each other within the storage system and to other objects outside the storage
16 system over the network with which they are operatively coupled. Also, the storage
17 system itself can be an object and interact as a node with other nodes in a network.

18
19 The relationship between and amongst these specific objects in the storage system
20 is usually visualized or characterized as a “tree” of objects. In a tree, each such object
21 hangs off a preceding object as if in a parent-child or inheritance relationship, with many
22 children hanging from a parent not being an atypical configuration. In addition to these
23 tangible kinds of objects, logical units (LUNs) are other nodes or objects that can be

1 contained within the tree. For example, a storage system object can have several LUN
2 objects as its children which, in turn, can have various disk objects as their children, etc.
3 These kinds of objects are generically referred to herein as “system objects” since they all
4 relate to a system or to components within a system, whether it is a storage system,
5 computer system, disk drive system, or some other system, and representations of these
6 objects can typically be displayed on a computer terminal’s graphical user interface
7 (GUI) in this tree fashion. However, in contrast, other kinds of objects (nodes) can also
8 be formulated which do not relate to a system or its components per se, such as objects
9 relating to user actions and represented on the GUI in other ways. (User actions are any
10 commands or operations initiated by the user, such as, for example, creating a LUN or
11 downloading new software to a disk, etc.). In addition, there can be yet other kinds of
12 network nodes beyond the two types mentioned such as nodes comprising or represented
13 by a communication tree.

14
15 Accordingly, it shall be appreciated that there can be a very large number of
16 nodes of various kinds or “personalities” to keep track of and manage within even a
17 relatively small and local computer network. Any one or more of these nodes can fail,
18 for one reason or another, either temporarily or permanently, and certain other nodes can
19 be added or removed by network users under certain conditions, and this complex and
20 dynamic network node scenario must be efficiently and effectively managed if the
21 computer network’s intended purpose is to be fulfilled. Moreover, this network node
22 management problem is compounded when the network is large, and possibly worldwide.
23 The subject of these computer network nodes is discussed further in two patent

1 applications filed by the assignee of the present invention: "Plug and Play Interface for
2 User Actions", Desai et al, U.S. Serial No. 09/916102, filed July 26, 2001 and "Scalable
3 Communication Within a Distributed System Using Dynamic Communication Trees",
4 Bauer et al, U.S. Serial No. 09/877862, filed June 8, 2001, both of which are incorporated
5 by reference herein in their respective entireties.

6
7 One prior art solution to the problem of managing network nodes is to elect one
8 node to have complete information about all other network nodes in the system.
9 Similarly, a group of nodes can be elected where each elected node in the group is
10 assigned to its own subnet or domain of nodes and has complete information about all
11 other nodes in its own subnet or domain. A subnet or domain is a network unto itself.
12 All other nodes in that network, or in that subnet as the case may be, then seek
13 information necessary to their functioning from their respective one elected node and are
14 thus managed through that one elected node. The inherent weakness in this configuration
15 is that if such one elected node fails, then no other node in its respective network or
16 subnet can function resulting in a failed network or subnet. This is a single point of
17 failure design which is not an optimum design because of at least this problem.

18
19 Prior art attempts to make this single point of failure design more reliable
20 included use of backup nodes, to take over functioning of an elected node if and when the
21 elected node failed. The problem with this backup design is that backup nodes can also
22 fail and then the result is the same as before – a failed network or subnet within the
23 network. Furthermore, when dealing with a worldwide network, which is not atypical

1 today, if backup nodes are being relied upon because of potential elected node failures,
2 and if all backup nodes are being maintained in one locale (e.g. United States of
3 America) for convenience, security, or other purposes, then virtually all network node
4 information can be destroyed for that global network if a disaster at the backup locale
5 destroys the backup nodes. This would be a major disruption. All worldwide users
6 (Europeans, Asians, etc.) of that worldwide network would thus lose the service provided
7 by that network.

8
9 Effective solutions to these single point of failure problems of the aforementioned
10 prior art are presented in the patent application cross referenced herein, and include use
11 of a master node which can be selected from the ranks of participating nodes in a network
12 domain or configuration. The master node is a privileged node relative to its
13 participating nodes and controls the directory database (DDB) distributed within each of
14 its participating nodes to match its own DDB and thereby be substantially identical from
15 node to node across its domain. The master node is selected or replaced with another
16 node under control of a global administrator, a privileged network user having special
17 password access to enable only him/her to perform these master node changes. It is
18 prudent computer network practice to have only one global administrator per domain, or
19 one person of authority under such global administrator per domain, who selects and
20 changes master nodes.

21
22 This master node solution is very satisfactory in any DDB network or domain
23 within that network in which only one master node asserts itself on participating nodes in

1 that network or domain, which may be the usual case, particularly with LANs. However,
2 if another user of the network gains access to the global administrator's password, either
3 with or without proper authorization, and if that other user selects a second master node
4 (a purported or contending master node) in the same domain that had a previously
5 operative master node, then a conflict results. All nodes in the configuration would then
6 be subject to assertions by two different master nodes that each of those master nodes is
7 the true master for each node in that domain. This problem may be compounded further
8 if there are more than only two users having this password access. This problem may be
9 compounded even further if this is a globally-dispersed network where human language
10 barriers can be challenging and result in multiple masters being created by mistake.
11 Further, widely-varying time zone differences (some users are sleeping while others on
12 opposite sides of the globe are working on the global network) might also contribute to
13 multiple masters being created by mistake.

14
15 Thus, there is need for a solution to this multiple master conflict problem, and the
16 present invention is an effective solution.

17 18 SUMMARY OF THE INVENTION

19 Embodiments of the present invention include apparatus, method, system, and/or
20 computer program product for resolving multiple master node conflict in a computer
21 network having a plurality of nodes only one which should be master node. The conflict
22 is between at least a first purported master node and a second purported master node.
23 One of these nodes is chosen to be master node.

1
2 In another aspect of the present invention a computer network has a plurality of
3 nodes each of which has a directory database (DDB) and one of which should be master
4 node. The master node is used to maintain contents of the DDB in each of the nodes
5 consistent throughout the plurality in a manner to avoid a single point of failure. A DDB
6 is a database of directory information, such as a directory of addresses of these nodes.
7 These nodes, such as computer storage systems, interact with computer network
8 information and can receive, store, modify and/or transmit the information.
9 Embodiments of the present invention include apparatus, method, system and/or
10 computer program product to resolve conflict between a first purported or contending
11 master node and a second purported or contending master node within the plurality of
12 nodes, by establishing a standard for comparison between the first purported master node
13 and the second purported master node, by comparing the first purported master node
14 against the second purported master node in accordance with the standard, and by
15 selecting either the first or second purported master node to be the master node based on
16 the comparison. In such apparatus there is a device that establishes the standard, another
17 device that compares the two purported masters, and yet another device that selects one
18 or the other purported master. The purported master node not selected is demoted to non-
19 master status and is returned to the plurality of nodes as a participating node in that
20 plurality.
21

22 In yet another aspect of the present invention, the standard is a temporal standard.
23 When the first purported master node was selected master node of the network a first

1 time of selection was obtained and when the second purported master node was selected
2 master node of the network a second time of selection was obtained. The first time and
3 second times of selection are compared to obtain the most recently selected purported
4 master node. The most recently selected purported master node is allowed to be master
5 node and the other purported node is demoted to non-master status as a participating node
6 in the plurality. If both purported master nodes are selected simultaneously, a default
7 condition results whereby the purported master node having the lower IP address
8 becomes the master node. (The higher IP address could have been used instead.)
9

10 In a further feature of the present invention, in the first purported master node, the
11 local time of selection of the first purported node as master node is recorded and the
12 duration of the selection is measured to obtain a first selection duration. The duration is
13 communicated to all other nodes in the plurality. Similarly, in the second purported
14 master node, the local time of selection of the second purported node as master node is
15 recorded and the duration of that selection is measured to obtain a second selection
16 duration. The second duration is also communicated to all other nodes in the plurality.
17

18 For each one of the nodes in the plurality except for the two purported master
19 nodes (for the slave nodes), the local time of receipt of the first selection duration is
20 noted and the first duration is subtracted from that local time of receipt to obtain first
21 adjusted local time, the local time of receipt of the second selection duration is noted and
22 that second duration is subtracted from that second local time of receipt to obtain second
23 adjusted local time, and the two adjusted local times are compared to obtain the most

1 recent adjusted local time. The node associated with the most recent adjusted local time,
2 either the first purported master node or the second purported master node, is identified in
3 each of the slave nodes.

4
5 Concurrently with this activity in the slave nodes, in the first purported master
6 node the local time of arrival of the second selection duration is noted and that duration is
7 subtracted from that local time to obtain a first purported master node adjusted
8 competitive local time. That competitive local time is compared with the first purported
9 master node's local time of selection as master to obtain a first most recent selection time.
10 The node associated with this first most recent selection time, either the first purported
11 master node or the second purported master node, is also identified.

12
13 And, concurrently with these activities in the first purported master node and
14 slave nodes, in the second purported master node the local time of arrival of the first
15 selection duration is noted and that duration is subtracted from that local time to obtain a
16 second purported master node adjusted competitive local time. That competitive local
17 time is compared with the second purported master node's local time of selection as
18 master to obtain a second most recent selection time. And the node associated with this
19 second most recent selection time, either the first purported master node or the second
20 purported master node, is also identified.

21
22 The number of times that the first purported master node is identified in each of
23 the slave and purported master nodes as the most recently selected master node is tallied

1 to obtain a first total, and the number of times that the second purported master node is
2 identified in each of the slave and purported master nodes as the most recently selected
3 master node is also tallied to obtain a second total. If the first and second totals are equal
4 the tie is broken by selecting the purported master node that has the lower IP address
5 (again, the higher address could have been used as selection criteria). But, if the totals
6 are different, the final master node selection is made by selecting whichever purported
7 master node had the greater total. The purported master node with the lower total is
8 demoted to non-master-node status and returned to the plurality of nodes as a
9 participating node.

10
11 It is thus advantageous to employ embodiments of the present invention in any
12 computer network, including client-server networks, whether a LAN or globally
13 dispersed WAN, where management of a distributed directory database is implemented
14 by way of a master-node to slave-node arrangement in which only one master node
15 should exist. For example, it is advantageous to employ embodiments of the present
16 invention in computer networks having a plurality of nodes or storage systems each of
17 which contains a DDB maintained consistent by a master node selected by a global
18 administrator from the plurality, where the maintenance is performed in a manner to
19 avoid a single point of failure and only one master node is permitted for that network at
20 any given time. Embodiments of the present invention ensure that only one master node
21 shall be permitted at any given time.

22

1 It is therefore a general object of the present invention to provide an improved
2 computer network.

3
4 It is another general object of the present invention to provide improved
5 distributed management software.

6
7 It is a further object of the present invention to provide an improved technique for
8 managing a directory database distributed over a plurality of nodes in a network such as a
9 client server network, such technique allowing only one master node at a time in the
10 plurality.

11
12 It is an even further object of the present invention to provide an improved
13 computer data storage system.

14
15 Other objects and advantages will be understood after referring to the detailed
16 description of the preferred embodiments and to the appended drawings wherein:

17
18 **BRIEF DESCRIPTION OF THE DRAWINGS**

19 Fig. 1 is a block diagram of a client server network having multiple servers and a
20 network node cloud in which embodiments of the present invention are particularly
21 useful;

22 Fig. 2 is a schematic diagram showing unconfigured nodes within a defined
23 boundary;

1 Fig. 3 is a schematic diagram showing configured nodes in three different
2 domains within the same defined boundary;

3 Fig. 4A is a schematic diagram of a network containing two domains of network
4 nodes having a master node in each domain;

5 Fig. 4B is a schematic diagram of the directory database distributed throughout
6 one domain in the network of Fig. 4A;

7 Fig. 5 is a flowchart reflecting an algorithm executed by operation of
8 embodiments of the present invention in establishing a configuration;

9 Fig. 6 is a flowchart reflecting an algorithm executed by operation of
10 embodiments of the present invention in performing a master to node handshake;

11 Fig. 6A is a flowchart reflecting an algorithm executed by operation of alternative
12 embodiments of the present invention in performing a master to node handshake;

13 Fig. 7 is a flowchart reflecting an algorithm executed by operation of
14 embodiments of the present invention in performing a node to master handshake;

15 Fig. 8 is a representation of a typical computer terminal screenshot associated
16 with certain embodiments of the present invention employing a GUI, the screenshot
17 reflecting a dialog box of the type which would enable a global administrator user to
18 control master node selection;

19 Fig. 9 is a representation of a typical computer terminal screenshot associated
20 with certain embodiments of the present invention employing a GUI, the screenshot
21 reflecting a dialog box of the type which would enable a global administrator user to
22 control domain configuration or re-configuration;

1 Fig. 10 is a schematic diagram of a global network in which a master to master
2 conflict can be resolved by embodiments of the present invention;

3 Fig. 11 is a flowchart reflecting part of an algorithm executed by operation of
4 embodiments of the present invention in resolving multiple master node conflict;

5 Fig. 12 is a flowchart reflecting a continuation of the algorithm of Fig. 11
6 executed by operation of embodiments of the present invention in resolving multiple
7 master node conflict;

8 Fig. 13 is a flowchart reflecting a continuation of the algorithm of Figs. 11 and 12
9 executed by operation of embodiments of the present invention in resolving multiple
10 master node conflict; and,

11 Fig. 14 is a flowchart reflecting the remainder of the algorithm of Fig. 11-13
12 executed by operation of embodiments of the present invention in resolving multiple
13 master node conflict.

14 **DESCRIPTION OF THE PREFERRED EMBODIMENTS**

15 Preliminarily, section headings which refer to figures to which they principally
16 relate are used hereinbelow as an organizational aid. However, there may be discussion
17 of subject matter reflected in a particular figure which appears in a section headed by a
18 figure number associated principally with other subject matter. The section headings are
19 thus not intended to be construed in a limiting manner. The terms "master node" and
20 "master" may be used interchangeably herein and have the same meaning. And, it should
21 be kept in mind that any network in which embodiments of the present invention are
22 operative has an intended purpose beyond management of its nodes. For example, in a
23 computer data storage system network, in which embodiments of the present invention

1 are particularly useful, the main purpose of each node or storage system in such network
2 is to enhance the overall network function of storing data, and management of its
3 distributed DDB is only one operation involved in that overall data storage function.
4

5 **Figure 1 – Client-Server Network**

6 Fig. 1 is a block diagram of an exemplary client server network having network
7 components such as multiple servers and a network node cloud. Embodiments of the
8 present invention are particularly useful in this network. User interface (UI) or client or
9 management workstation 100 is connected by bidirectional bus structures 106 and 107 to
10 servers 101, 102, and 103. UI 100 also is operatively coupled to network node cloud 105
11 by bidirectional bus structures 106, 107, 108. And server 103 is operatively coupled to
12 storage system 104 by bidirectional bus 109. The network is not limited to three servers
13 or to one storage system, and more than three servers are intended to be suggested by the
14 broken line representing bus 107 in connection with the dots between servers 102 and
15 103. In a particular configuration, UI 100 can be a Windows NT or UNIX workstation
16 that runs Navisphere 6.X user interface software or equivalents thereof; servers 101-103
17 can be any Windows NT machines or EMC model FC 4700 storage systems running
18 Navisphere 6.X management servers or equivalents thereof; and, storage system 104 can
19 be an EMC model FC 4700 storage system or its equivalent. Network node cloud 105 is
20 intended to represent a network such as a network of EMC FC-4700 storage systems or
21 their equivalents, containing a huge number of nodes, possibly thousands or more
22

1 As shown in Fig. 1, a directory data base (DDB) is situated in each network
2 component shown in the network. A DDB is a database of directory information
3 reflecting identities and addresses of both the node in which it resides and the other nodes
4 with which it communicates, much like a telephone directory book can list names, phone
5 numbers and addresses of all telephone subscribers to a telephone network. Since each
6 network component may include a large number of network nodes, as discussed earlier,
7 the DDB designation shown in each component is intended to represent DDBs distributed
8 throughout all nodes in that component. (The DDB shown in server 101, for example,
9 represents a plurality of DDBs.) In accordance with principles of the present invention to
10 be discussed in more detail below, under certain circumstances the DDBs in each
11 component are maintained consistently throughout the entire network. (The entire
12 network could be divided into independent subnets or domains, each being a sub-
13 network, described below.) In other words, the information content in any one particular
14 DDB in any particular node of any particular component in the network is maintained
15 consistent (replicated, duplicated, or made substantially identical) with content of any
16 other DDB in the network. Using the telephone book analogy, each telephone user
17 within that calling locale (the network) should be given the same revision or version of
18 the telephone book (the same DDB content – the same version number). Then, all such
19 users (nodes) have current information about all other telephone users listed in their
20 respective books. If a telephone user moves away, (node deleted) or if a new telephone
21 user moves into the locale (node added), then the telephone book needs to be updated
22 with the new information. The new telephone user needs to be given a complete and
23 current telephone book (current DDB). Under these conditions all telephone users in that

1 locale can rely on their respective telephone directories and can properly function.
2 Similarly, if the information content in each DDB in each node in the network is
3 maintained consistent regardless of network node changes (node additions, node
4 deletions, node failures, master node failures, failed links, etc.), then the computer
5 network nodes can rely on their respective database directories (DDBs) and can properly
6 function.

8 **Figures 2 & 3 – Nodes**

9 Fig. 2 is a schematic diagram showing unconfigured nodes within a defined
10 boundary. Bounded space 200 may represent, for example, a portion of a computer
11 system which has not yet been organized into an interrelated or cooperative entity, such
12 as may exist in a recently purchased system just delivered to the purchaser and prior to
13 such system being configured. The squares in the space represent nodes. The arrows in
14 the squares pointing in random directions are intended to suggest that these nodes are not
15 yet aligned with each other in a cooperative computer network effort. Each node's DDB
16 has only its own identification information in it – its own IP address. Thus, each node in
17 this scenario is functionally unaware of other nodes within this boundary and if it were to
18 respond to any computer information stimuli (data and/or commands) such response
19 would probably be independent of, uncooperative with, and/or contradictory to responses
20 to the same stimuli received by other unaligned nodes. Bounded space 200 is not a
21 network - yet.

Fig. 3 is a schematic diagram showing the same nodes as those in Fig. 2, but now shown as configured nodes in a network of three different domains, 301, 302, and 303 within the same defined boundary as that of Fig. 2. The same bounded space 200 thus contains the same nodes as before, but which have now been aligned, figuratively speaking, as shown by arrows, and segregated into domains within the same network boundary. (There is no literal, physical alignment or orientation of nodes into a particular geometrical direction – Fig. 3 reflects such orientation as an aid to comprehension.)

Alignment, in this context, means that a particular node is subordinate to, and takes its direction from, *only* its master node. In accordance with principles of the present invention to be discussed in more detail below, in domain 301 nodes are aligned to node M1, in domain 302 nodes are aligned to node M2, and in domain 303 nodes are aligned to node M3. Nodes M1, M2, and M3 are termed “master nodes”. The master nodes are nodes selected from the prior field of nodes within bounded space 200 and are made a little special or privileged as compared with the nodes now in alignment to them.

Aligned nodes are termed “participating nodes”. The responsibility of a master node is to ensure that contents of DDBs of all participating nodes in its configuration (all nodes in domain 301 including master M1 are in the configuration of master M1, etc.) are maintained consistent or substantially identical. The master node’s job is to replicate or duplicate the information content stored in its own DDB into each DDB in each participating node in its configuration. As noted, after each such DDB is replicated as desired it can be conceptualized as being “aligned” to its master node, as exemplified by the arrows superimposed on the nodes in domains 301, 302, and 303 pointing, respectively, in alignment to master nodes or masters M1, M2, and M3.

1
2 These master nodes are selected by a computer network user termed a “global
3 administrator”, a user who is privileged as compared with other users in that computer
4 network. Only the global administrator has proper authorization to select or change
5 master nodes, and make other network changes. Other users have rights limited to
6 merely using the network, as is. The global administrator selects nodes M1, M2, and M3
7 through a network user interface, e.g. UI 100 of Fig. 1, based on factors relevant and
8 important to the administrator. For example, selection of master nodes could be based on
9 proximate locations of such nodes to those intended to be in their respective
10 configurations. Another example would be selection of a master node based on its
11 functional relevance to the intended function of its configuration. The internet protocol
12 (IP) addresses of the nodes to be selected as master nodes are known to the global
13 administrator by way of the GUI, about which more shall be discussed hereinbelow.

14 15 **Figures 4A/B– Network & Domains - Overview**

16 Fig. 4A is a schematic diagram of an exemplary network containing two domains
17 of network nodes having a master node in each domain. This diagram shall be used in an
18 overview discussion involving creation of a configuration, and shall use the example of
19 adding nodes to the configuration as an introduction to certain concepts of the present
20 invention. More detail about these concepts shall be presented hereinbelow in connection
21 with other Figures.

1 Assume that network 400 initially was not configured as shown, but that it was
2 merely a boundary which contained nodes 1-8 inclusive in an unconfigured state, such as
3 reflected in Fig. 2. The first action undertaken is for the global administrator to pick a
4 node as master, and, in this example, node 1 is picked. Node 1 is selected by way of a
5 GUI (not shown in this Fig.) through a "select master" dialog box which appears on a
6 terminal screen of the GUI and which reflects IP addresses of nodes in this boundary. At
7 this point, the master node represents a configuration of one node - no other node in the
8 otherwise unconfigured boundary is aware that node 1 has been made master node.
9 Then, assume that the global administrator adds nodes 2 and 3 to the configuration, in
10 that order, by sequentially selecting such nodes' respective IP addresses from a
11 "configure domain" dialog box on the terminal screen of the GUI, by placing the terminal
12 screen cursor and clicking the mouse on "buttons" appearing in this dialog box which
13 cause these nodes to be sequentially added to the configuration. More about this
14 procedure will be presented in connection with Figs. 13 and 14 hereinbelow. At this
15 point, however, status of all nodes is: DDBs in nodes 1, 2, and 3 each have the same
16 contents, namely: IP addresses of nodes 1, 2, 3, and a flag that node 1 is master, and
17 nodes 4-8 remain unconfigured for the time being. Domain 401 has thus far developed
18 into a three-node configuration. The unconfigured nodes are further discussed
19 hereinbelow.

20
21 Backtracking momentarily, when the global administrator is adding node 2 to the
22 configuration, such node is necessarily being added through, or by way of, the only node
23 that pre-existed node 2 in the domain 401 configuration, namely node 1. Accordingly,

1 node 1, acting as master node, forwards information to just-arrived node 2 which advises
2 node 2 that node 1 is going to be master for node 2 and that node 2 has been added to the
3 configuration of node 1. Node 2 is thus a "master-added node". Node 2 then checks the
4 version number of the DDB in information received from the master node. A version
5 number is a unique worldwide number (WWN), a number that if once used is never used
6 again. In this example of a new configuration being created out of a boundary of chaotic
7 or unaligned nodes the version number of the master node's DDB could be zero. If the
8 version number in information sent by the master node is different from the pre-existing
9 version number of the DDB in node 2 (the likely case), then node 2 needs to get a copy of
10 the entire contents of the master's current DDB and to update (replace) the DDB in node
11 2 to be identical to the DDB in node 1. But, if version numbers of the DDBs in nodes 1
12 and 2 are the same (not likely), then the foregoing update/replacement is avoided. The
13 same basic procedure is undertaken for node 3 and for any other node to be added to the
14 domain 401 configuration, but there is some significant variation to the procedure as a
15 function of which node in the configuration is selected by the global administrator to be a
16 "portal node"- the node through which a new node is to be added. If the portal node is
17 master node, then the above procedure is used, but if the portal node is a configured node
18 within domain 401 other than master node then a different procedure is used, as follows.

19
20 If the global administrator adds node 3 through node 2 thus making node 2 a
21 portal node for new node 3, node 2 stores the IP address of node 3 in a cache memory
22 within node 2 (not within the DDB of node 2), sends a message to master node 1 that
23 node 3 has been or is being added and provides the IP address of node 3 in that message.

1 If master node 1 successfully receives this message from node 2, master node 1
2 immediately updates itself by adding node 3 IP address information to its DDB and
3 changes its version number. The version number changes from one unique WWN to a
4 different unique WWN. Concurrently, node 1 sends a "success note" back to portal node
5 2 about which more will be said below. Further, master node 1 sends a message to node
6 3 that the master for node 3 is node 1 and also provides node 3 with the changed new
7 version number of node 1 as well as a copy of the contents of its entire DDB. Newly
8 arrived node 3 accepts such contents, updates its DDB therewith, applies the changed
9 new version number to such contents, and is thereafter a participating node along with
10 node 2 in a configuration designated as domain 401 in which node 1 is master node.

11
12 Although, node 2 has retained the IP address of newly-added node 3 in its cache
13 memory, node 2 has not yet been updated. The earlier-mentioned success note contains
14 at least the master node's prior version number as well as its changed new version
15 number. Node 2 compares the master node's prior version number received in the
16 success note against its own version number of its DDB. On the one hand, if these
17 version numbers match, portal node 2 updates its own DDB with node 3 IP address
18 information which is being stored in the node 2 cache memory, thereby relieving the
19 master node from unnecessarily returning node 3 IP address information back to node 2.
20 Portal node 2 also updates its version number to be the same as the changed new version
21 number of the master node which was received in the success note. But, on the other
22 hand, if the prior version number of master node 1 did not match the version number of
23 portal node 2, then portal node 2 destroys contents in its cache memory about new node 3

1 and waits for its update message from master node 1. (The update message is a change
2 notification to all participating nodes in the master node's domain and is part of the
3 master node's replication process.) If there were many other operative and participating
4 nodes in domain 401, they would all be subjected to this replication process and portal
5 node 2 would be just one of all such participating nodes being updated with information
6 about newly added node 3 at the time of the update message. A significant aspect of this
7 procedure is that portal node 2 receives minimal communication from master node 1 to
8 achieve update of its DDB as noted above. Accordingly, any master node is worked
9 minimally and network or configuration "housekeeping" traffic is kept minimal in the
10 present invention while maintaining contents of distributed DDBs consistent throughout
11 any given node configuration.

12
13 Next, consider node 4, another new node being added to the configuration of
14 domain 401. If the global administrator, by way of the computer network's GUI, selects
15 node 3 as portal node for node 4, and if node 3 cannot properly communicate with master
16 node 1 for one of a number of reasons such as the network link between master and node
17 3 (not shown in Fig. 4A) has failed, then node 4 cannot be added to the configuration
18 through node 3. In such an event, an error message is sent to the GUI so that the global
19 administrator knows to select another node as portal node for node 4. The global
20 administrator can again select node 2 as portal node, this time for node 4, and node 4 can
21 thereby be successfully added to the domain 401 configuration in a manner similar to the
22 adding of node 3. (Another reason why node 4 might not be able to be added to the
23 configuration is that the master node itself has failed in which case an error message is

1 again returned to the global administrator as before, but which additionally advises of
2 failure of a master node. In this case the configuration is frozen and cannot be changed
3 until a new master is created by the global administrator, and discussion of this event is
4 presented hereinbelow.)

5
6 Domain 402 is shown with nodes 5, 6, 7 and 8. These nodes are shown in a
7 configuration with node 6 as its master node. These four nodes in this configuration have
8 been created in the same manner as those in domain 401, but either by a different global
9 administrator from the one in control of domain 401, or by the same global administrator
10 intending to establish a different and mutually- isolated domain from domain 401.

11 Accordingly, these two domains represent two different configurations which can co-
12 exist in a single computer network without either domain interfering with the other.

13 More than two domains can co-exist in a single network and only two are shown as an
14 example. (Related subjects are further discussed hereinbelow in connection with the
15 section on "Marriage of Two Subnets"). Nodes in this domain are shown to be
16 interconnected by communication lines 403-408 in a particular manner. More is said in
17 the incorporated-by-reference related application about these communication lines and
18 about master node "pinging" and participating node "polling" techniques utilized in
19 certain failure scenarios.

20
21 In Fig. 4B, the DDB which is distributed throughout the configuration of domain
22 401 is depicted in schematic format. Content of this DDB reflects successful addition of
23 node 4 to the domain 401 configuration and is the same content for each DDB within

1 each of nodes 1-4. This DDB associates each of the nodes in the configuration with its
2 respective IP address. These nodes could thus be localized, as in a LAN, or could be far-
3 flung and located around the globe being connected through the Internet. Node 1 has a
4 flag associated with it, shown in the diagram as an "M" to designate node 1 as the
5 configuration's master. Since each node in the domain 401 configuration has these
6 contents in its DDB, each node knows that node 1 is its master, because each DDB
7 contains this "M" flag next to or associated with the node 1 address content in its
8 database. And each node in the configuration also knows the IP address of every other
9 node (all other nodes) in the configuration. The DDB contains domain identification
10 information, as reflected by "DOMAIN 401" shown in the Figure. And, the DDB also
11 contains a version number as shown, "DDB VERSION #___", to reflect or identify the
12 status or version of its content, namely, participating node information as currently stored
13 in its DDB. In the case of master node 1, as noted above, it has the capacity to forward to
14 participating nodes in its configuration not only its prior DDB version number which was
15 stored in its DDB, but also can forward its changed update version number as stored in its
16 updated DDB, in the event of a network configuration status change. A schematic
17 drawing for the DDB distributed throughout the configuration of domain 402 would be
18 similar to the DDB for domain 401, but would have different content reflecting different
19 IP addresses and version number.

20
21 The procedure of adding a new node to a configuration, whether a master or non-
22 master node in the configuration is used as portal node, requires certain handshaking
23 protocols. These protocols shall be discussed in detail hereinbelow in connection with

1 flowcharts depicting their respective algorithms. This discussion of Fig. 4 has considered
2 the example of adding a new node in connection with a high level overview of certain
3 aspects of the present invention. Other changes to a configuration including node
4 deletion, node failure, master node failure, and master to node link failure are possible
5 and are considered in detail in the related application, incorporated-by-reference herein.
6

7 **Figure 5 – Establishing Configuration**

8 Fig. 5 is a flowchart reflecting an algorithm executed by operation of
9 embodiments of the present invention in establishing a configuration or domain. Initially,
10 in block 501 a chaotic array of nodes are defined within a boundary, such as those nodes
11 shown in Figure 2. The algorithmic process moves to block 502 wherein a node is made
12 master node by the global administrator through operation of the create master dialog box
13 in the GUI, thereby creating a configuration of one node. In this case, one of the nodes
14 within the boundary can be selected to be the master node. (A node outside of the
15 boundary could have been chosen by the global administrator to be the master node.)
16 Software, employed in preferred embodiments of the present invention, by which the
17 master is chosen and by which all other objectives of the present invention are
18 accomplished, is C++ object oriented software. Such software is the best mode now
19 known by which to implement the present invention, although other object oriented
20 software such as Java could have been chosen.
21

22 The algorithmic process then moves to block 503 where the global administrator
23 chooses other nodes through the GUI from either within or outside of the boundary to be

1 added to the configuration. Next, in block 504, the master node sends its IP address to all
2 the other nodes chosen to be within the configuration informing each of them that the
3 master node is their master. (Employing personification, in such “informing” or
4 “announcing” the master is effectively saying: “I am the master for you”.) This
5 informing is the aforementioned pinging and is also used in connection with certain
6 failure scenarios discussed in the incorporated-by-reference application. In block 505
7 each chosen node enters the master node’s IP address in its DDB, acknowledges the
8 master node, and thus aligns itself to the master node. This acknowledgment is vital for
9 the master to receive from each node in its configuration so that the master knows that
10 each such node is properly aligned. Failure to receive an acknowledgment from a
11 particular node alerts the master that such node might have failed and sets up a repetitive
12 pinging activity. The actions in blocks 503, 504, and 505 are accomplished by way of the
13 configure domain dialog box in the GUI. The steps or actions in this Figure are repeated
14 as many times as are necessary to accomplish the total number of domains desired per
15 network. Thus, in Fig. 3, within network boundary 200, three subnets or domains are
16 formed by these steps being undertaken three times by either one or multiple global
17 administrators.

19 **Figures 6 and 6A – Master to Node Handshake**

20 Fig. 6 is a flowchart depicting an algorithm executed by operation of
21 embodiments of the present invention in performing a master to node handshake. For
22 situations wherein a master node has need to send substantive information to its

1 participating nodes, this handshake is used with each such participating node prior to the
2 master sending that information to that participating node.

3
4 In query block 601, for each node in the configuration, a decision or
5 determination is made regarding whether or not the master node's IP address in the
6 configuration update (for either a node added or node removed update) matches the
7 purported master node's IP address stored in each node's DDB, as verification of the
8 master's identity. Verification is important because participating nodes in a configuration
9 must receive commands and updates from only their one true master node. Without
10 verification a problem could arise. For example, if a master node fails, requiring the
11 global administrator to select a different node as new master node, the new master node
12 sends out notification to all nodes in its configuration that it is their new master node. All
13 participating nodes in the configuration realign themselves to the new master node
14 without problem. But, if the failed master node recovers after selection of the new master
15 node, and if a user or different global administrator erroneously logs-in to that previously
16 failed master, there is a chance that it might be erroneously used as master in the same
17 configuration, which has to be prevented. The verification operation defined by query
18 block 601 prevents this problem from occurring

19
20 In block 601, if there is an IP address mismatch whereby the answer is "no" for
21 any particular participating node in the configuration, then in block 605 the change or
22 configuration update is not recognized by that particular node, the event is not logged
23 into the DDB of that particular node, and the handshake is concluded. In effect, this

1 means that the particular node sends a message back to the master node advising it that it
2 is not the particular node's master (in such case the likelihood is that the particular node
3 is participating in a different domain under a different master). Thereupon, the master
4 node removes the IP address of that particular node from the master node's DDB and
5 sends the change to all other nodes in the configuration. However, if the answer is "yes",
6 the process then moves to query block 602 where version numbers are compared. For
7 each participating node in the configuration, its DDB version number *before* the
8 configuration update is compared with the master node's DDB version number *before* the
9 update. This step ensures that each participating node's DDB contents were the same as
10 the master node's DDB contents before the update.

11
12 On the one hand, if the version numbers are the same whereby the answer to the
13 query in block 602 is "yes" then in block 604 only the update information (only the IP
14 address of the node being added or the node being removed) is accepted into that
15 participating node's DDB, and the handshake concludes. On the other hand, if the
16 answer is "no" then the version numbers do not match which means that contents of their
17 respective DDBs do not match. There is no readily available technique to determine to
18 what extent any such DDB may be out-of-date. Accordingly, for the situation of a
19 version number mismatch, in block 603 the participating node flushes its contents and
20 accepts the entire contents of the master node's DDB including both updated
21 configuration information about the added or removed node and a changed new master
22 node version number. Thereafter, the handshake concludes.

1 Referring next to Fig. 6A, a flowchart is depicted of an algorithm executed by
2 alternative embodiments of the present invention in performing a master to node
3 handshake. The flowchart is the same as that in Fig. 6 except for the addition of two
4 blocks, 606 and 607. If there is a mismatch between a master node's IP address as
5 reflected in an update message from that master node and second purported master
6 node's IP address as stored in the DDB of any participating node in the configuration,
7 then the algorithmic process moves to query block 606. In query block 606 a
8 determination is made regarding whether or not the other master node's IP address stored
9 in that participating node's DDB is null (invalid). On the one hand, if "no", not null
10 (meaning valid), then such address is in conflict with the master node IP address in the
11 update message, whereby the update/change is not recognized or accepted as reflected in
12 block 605, and the handshake is concluded. On the other hand, if "yes", meaning that
13 such address is null, then that nulled or invalidated address had earlier been made
14 ineffective and cannot possibly be in conflict with the master node IP address in the
15 update message. But the nulled address does indicate that such participating node's DDB
16 contents may be out-of-date. Accordingly, in block 607 such participating node flushes
17 the contents of its DDB, and updates its DDB with the new master name and a copy of all
18 contents stored in the master node's DDB, including both the update message and master
19 node DDB version number. Thereafter the handshake concludes.

21 **Figure 7 – Node to Master Handshake**

22 Fig. 7 is a flowchart depicting an algorithm executed by operation of
23 embodiments of the present invention in performing a node to master handshake. For

1 situations wherein participating nodes in a configuration have need to send substantive
2 information to their master node, this handshake is used for each such node prior to its
3 sending that information.

4
5 In block 701, an inquiring node having need to send substantive information to its
6 master gets the name and address of its presumed master stored in its own, local, DDB.
7 In decision block 702 that node inquires of the presumed master if it is that node's
8 master. If the presumed master node responds "yes", the handshake is concluded. But, if
9 the presumed master node responds "no", then the algorithmic process moves to another
10 decision block, block 703.

11
12 In block 703 the inquiring node inquires if the presumed master knows who is the
13 new master. If the answer is "yes", where the presumed master knows the identity of the
14 new master, the algorithmic process moves to block 704 wherein the presumed master
15 provides the name and IP address of the new master to the inquiring node after which the
16 handshake is concluded. But, if the answer is "no" the algorithmic process moves to yet
17 another decision block, block 705.

18
19 In block 705, which is needed when the presumed master does not know who is
20 the new master, a decision needs to be made about asking the global administrator to
21 configure a new master node by use of the select master dialog box in the GUI (discussed
22 in connection with Fig. 8). If the answer to the query in block 705 is "no", the inquiring
23 node inquires again in block 703 if the presumed master knows who is the new master –

1 it is possible that the presumed master may have learned of the new master's identity.
2 For example, the presumed master may have been a master that failed and was earlier
3 replaced with a new master by the global administrator. If the presumed (failed) master
4 was not brought up-to-date by the time of occurrence of the first inquiry by the inquiring
5 node, but is brought up-to-date at a later time whereby identity of the new master is then
6 made known to the presumed master, then inquiry of the presumed master (in block 703)
7 after that later time will result in provision of the new master's identity. This iterative
8 loop or iteration means thus operates until either block 704 is utilized as described above,
9 or until the decision in block 705 is "yes" – to ask the global administrator to configure a
10 new master node by way of the GUI. If "yes", the algorithmic process moves to block
11 706 wherein the global administrator configures new master node information using the
12 GUI which results in a new master for the inquiring node and the handshake is
13 concluded.

14 15 **Figure 8 – Select Master Dialog Box**

16 Fig. 8 is a representation of a typical computer terminal screenshot associated
17 with operation of certain embodiments of the present invention employing a graphical
18 user interface (GUI) with a computer network including storage systems. A GUI could
19 be employed within client or management workstation or user interface 100 in Fig. 1.
20 The screenshot depicts a dialog box of the type which would enable a global
21 administrator (privileged user) to control master node selection.
22

1 With regard to screen layout, in the upper left hand corner of the Figure, a “Select
2 Master” title in the toolbar is shown and in the upper right hand corner in that toolbar an
3 exit button “X” is shown and used to exit the dialog box. Under the heading “Master
4 Node IP Address” there is shown an editable field containing, in this example, a specific
5 IP address: 128.221.34.187. The global administrator can select or change master nodes
6 by editing this field, described below. There is a “Master Candidate Nodes” section in
7 the screenshot. This section displays nodes which are participating, configured nodes in
8 the domain and are thus available as candidates from which one can be selected or
9 elected by the global administrator as master. In this section, three categories of
10 information are provided: “Node Information”, “System Name” and Provider
11 Information”.

- 12 • Node Information includes IP address and system identifier of a particular
13 node or system. As noted earlier, an entire storage system can be treated in
14 object-oriented software as an object or node and thus can have an IP address
15 (or possibly more than one address) associated with it. The system identifier
16 is constructed from the last two segments of the IP address of that system (e.g.
17 128.221.34.187 is the IP address corresponding to system identifier
18 CPC34187).
- 19 • System Name is the name assigned to the system or node and is just a label or
20 tag.
- 21 • Provider Information includes name and version number of providers installed
22 in the system. A provider is a functional software module. This version
23 number is not the same as version number of the master node and its

1 participating nodes earlier discussed, but is version number of the provider
2 software.

3 There are three entries shown in the Master Candidate Nodes section, having system
4 names "Array 3", "Array 2" and "Array 1", with associated Node Information and
5 Provider Information. In the lower right corner of the Figure are four control buttons:
6 "OK", "Apply", "Cancel", and "Help".
7

8 With regard to operation, the global administrator can point the cursor and mouse-
9 click on Apply (can hit Apply button) to make a node selected from this group of three
10 nodes appear in the editable field above and become new master for this configuration.
11 In this example, the second candidate in the Master Candidate Nodes section
12 corresponding to Array 2 is selected as master node, as is reflected in the editable field.
13 The OK button performs the same action as the Apply button, but also closes the dialog
14 box. The Cancel button closes the dialog box without saving the global administrator's
15 changes thereby maintaining the status quo. And, the "Help" button launches a help
16 menu for this Select Master dialog box.
17

18 However, before any of this operation can be performed, the global administrator
19 first has to bring up the dialog box. And, before the dialog box can be brought up, the
20 global administrator first has to bring up a web browser on his/her terminal screen and
21 type into the Uniform Resource Locator (URL) slot the IP address of a storage system
22 chosen by the global administrator (in this example, one of the IP addresses for Array 1,
23 2, or 3). Such storage system may be part of a small or large domain or network of

1 storage systems and possibly other nodes. In this example the global administrator could
2 have typed-in the IP address of any of the three arrays shown and the same substantive
3 information would have been displayed because the DDB in each of the three storage
4 systems contains the same directory data. However, the information may be displayed in
5 a different order as a function of which IP address was typed in. By accessing a system
6 or node in this manner, a framework software is brought into operation, which provides
7 the framework or foundation of the terminal screen display. With reference to one of the
8 incorporated-by-reference patent applications, "Plug and Play Interface for User
9 Actions", Desai et al, U.S. Serial No. 09/916102, filed July 26, 2001, a dialog box of this
10 kind can be created by a software module that plugs into such framework software as
11 described in that patent application. However, after framework software comes up, a
12 security operation involving password access is next undertaken to ensure that only a user
13 with proper password (presumably global administrator) has access to the Select Master
14 dialog box. If the proper password is supplied, the framework software thereafter offers a
15 menu item "Select Master". And, if that menu item is pointed-to and clicked-on by the
16 global administrator, or other authorized user, then, this dialog box is finally launched.

17
18 The act of launching this dialog box causes reading of directory information from
19 the DDB in the storage system node chosen by the global administrator. In this example,
20 that DDB contains node, system name, and provider information of Array 3, Array 2 and
21 Array 1 respectively, and displays it as shown. This information shows that this DDB
22 has three storage arrays - three IP addresses in this case - and that the chosen storage
23 system is thus participating in a domain or configuration that has three nodes. In the case

1 where a storage array has two IP addresses associated with it, the same system name will
2 appear on the screen twice, reflecting both IP addresses on two separate rows.

3
4 Any one of these three nodes (if all are operative) can be selected by the global
5 administrator to be master node by merely highlighting that selection in the Master Node
6 Candidates section and hitting the OK or Apply button. This action will move the IP
7 address corresponding to that selection to the Master Node IP Address editable field and
8 replace the IP address of any node that might have been in that field. That replaced
9 master node is then demoted to being a participating node of the configuration.

10 (Referring back to Fig. 4B, the "M" identifier is moved from Node 1 to either Nodes 2, 3,
11 or 4 in that domain.) If one of the listed nodes had failed, for example Array 1, and is
12 thereafter selected to be master node, then after highlighting Array 1 and hitting the
13 Apply button the user interface returns an error signal indicating that CPC 3463 was not
14 able to be contacted. Another node will then have to be selected. As noted, in this
15 example, the global administrator selected array 2 and its IP address appears in the
16 editable field at the upper portion of the screenshot. After the global administrator is
17 finished with his/her master node selection, the dialog box can be closed out by hitting
18 the X in the upper right corner.

20 **Figure 9 – Configure Domain Dialog Box**

21 Fig. 9 is a representation of a typical computer terminal screenshot associated
22 with operation of the same embodiments of the present invention that produced the dialog
23 of Fig. 8. Fig. 9 depicts a dialog box of the type which would enable a global

1 administrator user to configure or reconfigure a domain. In the main toolbar section,
2 “Configure Domain” is shown as the title of the dialog box . Also, in the main toolbar,
3 right hand side, an exit button “X” is shown and is used to exit the dialog box by clicking
4 on it.

5
6 With regard to screen layout, there are three sections displayed: “Domain Name”
7 near the top, “Scan Subnets” below that, and “Select Systems” at the bottom half of the
8 screenshot.

- 9 • In Domain Name section its editable field shows name of domain of the
10 system selected by the global administrator. The term “DefaultDomain” is
11 shown in the field, and a default domain is automatically selected if not
12 overridden by a different selection made by the global administrator. There is
13 a “Change” button located to the right of this field.
- 14 • In Scan Subnets section there are two fields, “Subnets To Add” which is an
15 edit control and “Subnets To Scan” which is a list control. There are also four
16 buttons shown: “A”, “Scan”, “Stop Scan” and “Clear”. And there is a scan
17 progress bar.
- 18 • In Select Systems section there is an editable field entitled “IP Address of
19 System” operated upon by a “B” button to its right; an “Available Systems”
20 field operated upon by “C” and “D” buttons to its right, and a “Selected
21 Systems” field operated upon by three “B”, “C”, and “D” buttons to its left.
22 In addition, there is a “Clear” button at bottom of this section, as well as scroll
23 bars shown at bottom of both Available Systems and Selected Systems fields.

1 At bottom of the screenshot are four buttons: "OK", "Apply" "Cancel" and "Help".

2
3 With regard to operation, this dialog is brought up in the same manner as
4 described earlier in connection with the dialog of Fig. 8. In this case, there is a
5 "Configure Domain" menu item presented which the Global Administrator clicks on and
6 which launches this dialog box. When this dialog is launched, the name of the system or
7 node pointed to by way of inserting its IP address into the URL slot appears in the
8 Domain Name field in this dialog. If the global administrator wants to change this name
9 he/she hits the Change button which brings up another dialog (not shown) with an
10 editable field allowing the typing-in of a different name. The domain name is visible
11 from all participating systems in that domain which means that this same name would
12 appear in the Domain Name field regardless of which participating system's IP address
13 was put into the URL slot by the global administrator. Two domains can be given the
14 same name without malfunction, (they could all have the default domain name) but this
15 could lead to confusion. In any event, any domain name must be in alphanumeric form
16 only, and non-alphanumeric symbols are not permitted.

17
18 In the Subnets to Add edit control the global administrator types in the address of
19 a subnet which he/she wishes to have scanned to discover any available systems (systems
20 unaligned to any master node) that are also compatible with other nodes in the domain
21 which the global administrator is in the process of configuring or re-configuring. For
22 example, nodes or storage systems or servers which fall into the category of Common
23 Information Model Object Manager (CIMOM) systems or servers are mutually

1 compatible. Note that this address is not a full four-segment IP address, but is a subnet
2 address or identifier (ID) having only three segments. The “A” button located to the right
3 of the Subnets To Add edit control is used to for moving contents of that edit control to
4 the Subnets to Scan list control located to the right of the button, thereby populating that
5 list control field which is initially empty when the dialog is brought up. A populated list
6 control field thus reflects certain subnets which the global administrator wishes to scan.
7 As can be seen, in this example subnet address “10.14.12” is the last subnet that was
8 added to the Subnets to Scan field, as it is located at bottom of three subnet addresses
9 shown in that field. The global administrator uses the screen cursor to highlight one or
10 more subnets listed and hits the Scan button to the right of the list control to start a
11 discovery operation on those highlighted subnets. The discovery operation will cause a
12 display in the progress bar to show percentage of completion of scan operation. The Scan
13 button is disabled from time that scanning is started until it is finished. The Clear button
14 is used for clearing contents of list control, this button being enabled only if the global
15 administrator selects (highlights) a subnet listed in the list control. The Clear button is
16 disabled and a subnet cannot be highlighted in the list control field when scan operation
17 is in progress. The Stop Scan button is used to stop scan operation which was started by
18 hitting the Scan button, and will be enabled only if there is a scan in progress.

19
20 After a scanning operation has completed, where progress bar shows 100%, IP
21 addresses of any discovered nodes that are unaligned to second purported master node
22 and which are compatible with nodes in the domain under configuration are automatically
23 sent to the Available Systems field in the Select Systems section of the dialog. In other

1 words, these particular discovered nodes populate the Available Systems field which has
2 sub headings of "System" (storage system name) and "Node Info" (storage system IP
3 address). In the example shown, in the Subnets To Scan field, the subnet having address
4 128.221.34 contained an unaligned and compatible node with IP address 128.221.34.80
5 with system name Cadsys, and this is the first entry in the Available Systems window.
6 Likewise, the subnet having address 128.221.42 contained three unaligned and
7 compatible nodes: 128.221.42.64; 128.221.42.80; and, 128.221.42. 122 which are three
8 entries under Node Info in the Available Systems field. Finally, the last subnet scanned
9 having address 10.14.12 contained an unaligned and compatible node with IP address of
10 10.14.12.105 which is the last entry shown in the Available Systems field.

11
12 The global administrator now has a choice of selecting any or all of these
13 available systems for addition to the domain being configured. For example, to add the
14 node named Cadsys the global administrator would highlight this entry in the Available
15 Systems field and hit the C button. This action would move that Cadsys entry from the
16 Available Systems field which would thus be decreased by one entry, to the Selected
17 Systems field to the right of the button which would thus be increased by one entry.
18 Then the global administrator hits the Apply or OK button to cause the master node to
19 replicate the node added change in the DDBs of all nodes in the domain and thereby keep
20 the DDBs consistent throughout the domain.

21
22 In the Selected Systems field there are shown three nodes having the same IP
23 addresses that were shown in Fig. 8. These addresses appear in this field when this

1 Configure Domain dialog is brought up, which means that the domain under
2 configuration in Fig. 9 is the same domain for which the global administrator selected
3 Array 2 having IP address 128.221.34.187 as master node in Fig. 8. In other words, in
4 this illustrative example, the three nodes appearing in the Selected Systems field had been
5 earlier selected for this domain and can thus be considered as “pre-selected” since they
6 existed in the field prior to bringing up the dialog shown in Fig. 9. In accordance with
7 our example, this entry of three nodes in the Selected Systems field can be increased by
8 the addition of the Cadsys system and any others selected by the global administrator.

9
10 In the reverse operation, the population of the Selected Systems field can be
11 reduced by the global administrator highlighting an entry in the Selected Systems field
12 and hitting button “D” which moves that highlighted entry into the Available Systems
13 field. Then the global administrator hits the Apply button which causes the DDB in the
14 removed node to destruct (to discard its contents), thereby unaligning its DDB from
15 alignment to master node cpc34187. This unalignment makes the removed node
16 available for use by other global administrators. Hitting the Apply button also causes the
17 master node to replicate this node removal change in all the remaining DDBs in the other
18 nodes of this domain and thereby keep the DDBs consistent throughout the domain. The
19 “Clear” button is used to clear or empty contents of the Available Systems field to make
20 it open for repopulation by newly-discovered available systems derived from the
21 scanning operation in the Scan Subnets section of the dialog. The cleared systems which
22 are thus removed from view from the Available Systems field are otherwise unaffected
23 and their DDBs remain intact. Scroll bars at bottom of both the Available Systems field

1 and the Selected Systems field, also operable through left and right pointing arrow
2 buttons at lower left and right corners of the fields, permit vertical scrolling of entries in
3 both fields.

4
5 In the editable "IP Address of System" field located above the Available Systems
6 field the IP address 128.221.34.16 is shown. This is an IP address of a node about which
7 the global administrator has prior knowledge, and knows is both unaligned to any other
8 master and compatible with nodes of this domain under configuration. The global
9 administrator has thus typed-in this IP address into this editable field. By clicking on the
10 "B" button, this IP address is moved into the selected Systems field and the OK or Apply
11 button is then hit to add the node with this IP address to the configuration as described
12 above. If the global administrator later wants to remove it, highlighting it and hitting the
13 D button moves it over to the Available Systems field, in accordance with operation
14 discussed above.

15
16 It should be understood that embodiments of the present invention relating to the
17 GUI discussion of Figs. 8 and 9 utilize JAVA® object oriented software, including the
18 framework software discussed above and further discussed in the incorporated-by-
19 reference patent application cited in discussion of Fig. 8. Software of the present
20 invention operates on, through, and/or with a client workstation such as a Windows NT
21 or UNIX® workstation to accomplish various GUI functions and operations, including
22 those discussed in connection with both Figs. 8 and 9.

Figures 4A and 6 – Marriage of Two Subnets

With the GUI operational description of Figs. 13 and 14 in mind, and referring back to Fig. 4A and Fig. 8, consider the marriage of two subnets. In Fig. 4A, where two domains or subnets are shown operating independently, each subnet is shown to have its own master. The DDB of each master node contains an IP address for each node in its own subnet. These two subnets can be physically linked or joined together (married) by the global administrator operating through the GUI, so that any node from either domain can communicate with any node from the other domain. In this married state, although communicatively linked, the nodes from each subnet behave in a manner consistent with their prior independent subnet status. There is no master to master conflict in this situation because each master is not asserting itself (not saying: "I am the master for you") upon nodes from the other subnet.

Using Fig. 4A as an example, in a subnet marriage, the two masters (nodes 1 and 6) remain within their respective domains, and the domain boundaries around domains 401 and 402 remain intact. However, there will be an operative coupling, a communication link or bus (not shown), that connects nodes 1 – 4 inclusive from domain 401 to nodes 5 - 8 inclusive from domain 402. For example, node 3 could talk to node 8, etc. Accordingly, if an IP address from one of the nodes from one of the subnets, for example node 2 from domain 1, is inserted into a web browser's URL slot on the GUI terminal screen, and a scan operation of this marriage of two subnets is performed, then nodes from both subnets will be discovered. (This instant scan is similar to, but different from, that shown in Fig. 13. The scan of Fig. 13 was for the purpose of discovering

1 available nodes outside of a domain with which to configure or re-configure that domain,
2 but this instant scan is to discover nodes currently existing within a domain and is
3 performed through another dialog, not shown).

4
5 The result of this instant scan will cause node 2 to initiate an update to its master
6 node 1, advising it of new nodes in the domain. Master node 1 will update its DDB with
7 all node information from all nodes in both subnets. But, when master node 1 propagates
8 its update to all nodes from both subnets, those nodes which comprise domain 402 reject
9 the update. Those nodes have their own master node, master node 6. Because of the
10 master to node handshake, nodes 5, 6, 7, and 8 will reject this update request since the IP
11 address of master node 6 differs from the IP address of master node 1 (see Fig. 8, block
12 605). The global administrator will be able to see this at the GUI by looking at an
13 eventlog. Master node 1 will mark nodes 5, 6, 7, and 8 in its DDB as remote nodes
14 within its subnet. Likewise, master node 6 will mark in its DDB nodes 1, 2, 3, and 4 as
15 remote nodes within its subnet. Thereafter, until further intervention by the global
16 administrator, both sets of nodes co-exist in a marriage of subnets while retaining their
17 individual autonomy.

18
19 This network configuration has practical utility when a link between domains is
20 useful or economically feasible, but not on a continuous basis. For one example,
21 consider the case where a startup company on a tight budget has two offices located on
22 opposite coasts of the United States, and where they have need to access each others files.
23 This business arrangement might arise when a West coast project is started at a different

1 time from the East coast project and where it is useful to keep the two projects segregated
2 except for intra-company access to each other's files. This company may lease network
3 lines that are charged at an hourly rate to connect these two subnets cross country. And,
4 with this marriage scheme, they can be disconnected each evening and reconnected each
5 morning to effect a substantial savings in leased line costs. While these subnets are
6 connected, co-workers at opposite ends of the country can read each others files at great
7 convenience to the company, but they cannot otherwise impact each other's files, keeping
8 the two projects segregated. The DDBs in the East coast's nodes are isolated from the
9 West coast's master node and vice versa.

Figure 10 – Master to Master Conflict

10
11
12 Fig. 10 is a schematic diagram of a global network in which a master node to
13 master node conflict is shown to exist and which can be resolved by operation of
14 embodiments of the present invention. Worldwide network 1000 can be part of the
15 Internet network and nodes 1 through 16 are shown and can be located at any place on
16 the globe. For example, node 3 in the upper right hand corner of the diagram is shown to
17 be located in the UK and node 4 is shown to be located in the USA. Master node M-1 is
18 shown to be located in Japan and master node M-2 is shown to be located in Brazil. The
19 global administrator is shown to be located in the USA and an unauthorized user is
20 shown to be located in Japan. Places of location of each of these nodes and persons is not
21 important to operation of the present invention. And, the present invention will be
22 equally effective if the network was a LAN with few or many nodes and not global in
23 size. Although communication links are not shown between and amongst all depicted

1 nodes to enhance clarity of presentation, it is to be understood that appropriate links do
2 exist. These links, if part of the Internet, could be comprised of any or all of hard wire,
3 optical, telecommunication-via-satellite, and other link components.

4
5 In the example shown, a global administrator located in the USA properly selects
6 a node located in Brazil to be master node of this global network. This selection is made
7 via communication link 1005 by way of the dialog of Fig. 8 through the GUI located with
8 the global administrator in the USA, as earlier described. Thereupon, master node M-2
9 properly announces itself to nodes in its configuration which go through appropriate
10 acknowledgments and handshakes as described hereinabove, after which the
11 configuration (global network) runs properly.

12
13 The global administrator, as noted above, is a network user with special
14 privileges. Only this person, or someone under his/her authority, has appropriate
15 password access to the dialog of Fig. 8 to select or appoint a master node. If another
16 user, either subsequently or simultaneously, also selects a different node to be master
17 node, then a conflict results. In this example, a user located in Japan selects a node
18 located in Japan (M-1) to be master node, and this selection is made by way of
19 communication link 1004 through a GUI located in Japan with the unauthorized user. At
20 this point, M-1 announces itself ("I am the master for you") to each of the nodes in the
21 network and the nodes in the network are in conflict because they had been properly
22 aligned to master node M-2 and do not know if they should acknowledge new master
23 node M-1 or not. At this point there is a question or ambiguity about which node is really

1 the master node, whereby both nodes can now be referred to as “purported master” or
2 “contending master” nodes until such ambiguity is removed. However, for purposes of
3 facilitating discussion in connection with Figs. 10 - 14, “master” may be used rather than
4 “purported master” in every instance, but a purported master node under a cloud of
5 ambiguity shall be intended until the master conflict is resolved.

6
7 When master node M-2 had been selected, in addition to its announcement as
8 master, it had recorded its local Brazilian time of selection (LTS) as master and
9 thereupon starts recording its time duration as master node. Analogously and figuratively
10 speaking, master node M-2 starts its “stop watch” when it is selected as this network’s
11 master. Likewise, when unauthorized master node M-1 is selected, in addition to its
12 announcement as master, it records its local Japanese time of selection as master and
13 thereupon starts recording its time duration as master node. In addition to the
14 announcements made to the other nodes in the network by these masters at the times of
15 their respective selections (appointments), each also transmits its respective time duration
16 as master node to all other nodes in the network. These transmissions of time duration
17 are not necessarily coincident with the announcements, and any time delay from time of
18 selection until transmission of the time duration is accounted for in the transmitted time
19 duration value itself.

20
21 For example, consider node 4 which is shown with links to both master nodes.
22 Brazilian Master node M-2 sends its time duration value to node 4 in the USA on
23 transmission link 1002 and node 4 subtracts that value from its own local time to achieve

1 an adjusted local time of selection of M-2 as master node (shown as NODE 4's ADJ.
2 LTS -2). Node 4 records this LTS - 2 time. Similarly, Japanese master node M-1 sends
3 its time duration value to node 4 in the USA on transmission link 1001 and node 4 also
4 subtracts that value from its own local time to achieve an adjusted local time of selection
5 of M-1 as master node (shown as NODE 4's ADJ. LTS - 1). Again, node 4 records this
6 LTS - 1 time. As noted, this process (this time duration transmission and subtraction
7 from local time process) involves all nodes in the network. Accordingly, each master
8 node transmits to the other master node over transmission link 1003 its respective time
9 duration. At node M-1, it receives M-2's time duration which is subtracted from local
10 Japanese time to become Node M-1's adjusted LTS-2 which is stored in M-1. Likewise,
11 at node M-2, it receives M-1's time duration which is subtracted from local Brazilian
12 time to become node M-2's adjusted LTS - 1 which is stored in M-2. Each node in the
13 global network is thus supplied with temporal information with which it can make a
14 meaningful comparison in its respective local time zone between the absolute time of
15 selection of M-1 and the absolute time of selection of M-2. The detail with respect to this
16 comparison is provided in connection with discussion of flowcharts in Figs. 11 - 14
17 hereinbelow.

19 **Figures 11 - 14 Master to Master Conflict Flowcharts**

20 Fig. 11 - 14 are flowcharts which depict an algorithm implemented by
21 embodiments of the present invention in resolving master to master conflict. Referring to
22 Fig. 11, in block 1101, a local or worldwide configuration of nodes is managed by its
23 current master node, such as node M-2 in Fig. 10. In block 1102, a node which is not the

1 current master is selected by a network user, resulting in two purported master nodes,
2 node 1 and node 2 at the same time, such as master nodes M-1 and M-2 in Fig. 10. In
3 block 1103, node 1 and node 2 each records its respective local time of selection (LTS) as
4 master node at that respective time and concurrently therewith begins measuring its
5 respective time duration as master. In block 1104, node 1 and node 2 each asserts its
6 master status ("I am the master for you") on the other nodes in the configuration and
7 upon each other when each is selected. Conflict in each of the nodes in the configuration
8 occurs when it receives such an assertion or announcement either while aligned to its
9 master or while it is acting as master. Also, node 1 and node 2 each starts its respective
10 "stop watch" at the time it is selected master to measure its respective time duration as
11 master. In Fig. 10 these announcements are made on transmission links 1001, 1002, 1003,
12 and others (not shown) as earlier discussed. Thereafter, or simultaneously therewith, in
13 block 1105, each transmits its time duration to other nodes in the configuration and to
14 each other on those same links. In block 1106, each node upon receipt of a time duration
15 transmission subtracts such duration from its local time thereby obtaining and storing an
16 adjusted local time of selection (adjusted LTS) of a master node. In a conflict situation as
17 shown here, slave nodes (nodes other than purported master nodes 1 and 2) receive two
18 time duration transmissions and therefore obtain and store two adjusted LTS values,
19 where master nodes 1 and 2 each receive only one time duration transmission from the
20 other master node resulting in obtaining and storing only one adjusted LTS value. The
21 algorithmic process moves from block 1106 via tab "A" to Fig 12.

22

1 In Fig. 12, query block 1201 asks: does each node other than purported master
2 nodes 1 & 2 determine that its adjusted LTS-1 (that node's adjusted local time of
3 selection of master node 1) equals its adjusted LTS-2 (that same node's adjusted local
4 time of selection of master node 2). In addition, the question is also asked of each master
5 node: is its own LTS (each purported master's own local time of selection of itself- local
6 Brazilian time for M-2 and local Japanese time for M-1 in Fig. 10) equal to its adjusted
7 LTS of the other master node. In other words, for every node in this configuration,
8 including both master nodes, does each node arrive at the very same conclusion, namely,
9 that the two different selections of the two different master nodes occurred absolutely
10 simultaneously. In still other words, does each node in the network come to the very
11 same conclusion that two selections of two different nodes to be master node of the
12 network occurred within the same global network system clock pulse? [The probability
13 of this event occurring is very low. Even if such singular clock pulse event did occur,
14 variations in transmission time based on transmission link differences from node to node
15 between the purported master nodes and various slave nodes would be likely to cause at
16 least one node to have an opposite conclusion.] On the one hand, if "yes", then a default
17 condition occurs and the algorithmic process moves to question block 1202 where the
18 question is posed: which master has the lower IP address? If node 1's address is less than
19 node 2's address, the algorithmic process moves to block 1203 and node 1 is master. If
20 node 2's address is less than node 1's address, the algorithmic process moves to block
21 1204 and node 2 is master. Whichever node is not made master returns to participating
22 node status in the configuration. Since every IP address is a unique number, then such a
23 tie-event, although remote, can certainly be broken by these steps. The algorithmic

1 process then moves to block 1205 to bring in any handshaking that may be required after
2 which the process is concluded and the network is again operating under one master.
3 But, on the other hand, if even only one node disagrees on the question of simultaneity in
4 block 1201, then the algorithmic process moves via tab "B" to Fig. 13.

5
6 In Fig. 13, tab "B" has input to three question blocks. Question block 1301
7 handles nodes other than purported master nodes. Question block 1302 handles
8 purported master node 1, and question block 1303 handles purported master node 2. In
9 block 1301, for each node other than the two master nodes, the question is posed: is its
10 adjusted LTS for master node 1 more recent than its adjusted LTS for master node 2?
11 Each slave node that provides a "yes" answer is accounted for in block 1305 which
12 provides a total of yes answers. And, each slave node that provides a "no" answer is
13 accounted for in block 1304 which provides a total of no answers. In block 1302, the
14 question is posed: is master node 1's LTS of itself more recent than master node 1's
15 adjusted LTS of master node 2 (after node 2's time duration has been taken into account).
16 If "yes" that adds to the total number of yes answers in block 1305, but if "no" that adds
17 to the total number of no answers in block 1304. Finally, in block 1303, the question is
18 posed: is master node 2's LTS of itself more recent than master node 2's adjusted LTS of
19 master node 1 (after node 1's time duration has been taken into account). If "yes" that
20 adds to the number of *no* answers in block 1304, but if "no" that adds to the number of
21 *yes* answers in block 1305. In this manner each node in the configuration contributes to
22 either the total number of nodes which measured purported master node 1 as being more
23 recently selected as master than purported master node 2 or contributes to the total

1 number of nodes which measured purported master node 2 as being more recently
2 selected as master than purported master node 1. It should be understood that in the
3 normal course of operation of a network of this type, if there is a master to master
4 conflict, the overwhelming likelihood is that the two masters would have been selected at
5 times that were vastly different from each other – minutes, hours, days, weeks, months,
6 or longer times could separate the two times of selection. Under these more ordinary
7 conditions, output from block 1305 would total “yes 100%” while output from block
8 1304 would total “no 0%”, or vice versa. The totals from blocks 1304 and 1305 are
9 directed to tab “C” which connects to the input of Fig. 14.

10
11 In Fig. 14, totals from blocks 1304 and 1305 are compared in block 1401 and the
12 result is input to question block 1402. On the one hand, in block 1402, if there is a tie
13 yes/no count, (again, an extremely unlikely event) the algorithmic process moves to
14 question block 1406 where, as before, IP addresses of both purported master nodes are
15 compared to determine the purported master node having the lower of the two IP
16 addresses, and that purported master node is made master in blocks 1407 or 1408.
17 (Alternatively, the higher IP address could have been used.) On the other hand, if there is
18 no tie count, the algorithmic process moves to question block 1403 to determine if the
19 total number of nodes in the configuration that measure purported master node 1’s time
20 of selection as master being more recent than purported master node 2’s time of selection,
21 is greater or less than the total number of nodes in the configuration that measure
22 purported master node 2’s time of selection as master being more recent than purported
23 master node 1’s time of selection. If node 1 has a greater total, then node 1 is made

1 master in block 1405, but if node 2 has a greater total, then node 2 is made master in
2 block 1404. Thereafter, whichever result is obtained from either blocks 1404, 1405,
3 1407, or 1408, that result is subjected to handshaking in block 1409 in accordance with
4 Figs. 6 and/or 7, after which the algorithmic process stops. It should be understood that if
5 more than two masters are in conflict, the above procedure can be used to resolve conflict
6 between two of them at a time in accordance with the foregoing description until the
7 conflict is completely resolved.

8
9 The present embodiments are to be considered in all respects as illustrative and
10 not restrictive. The flowcharts used herein to demonstrate various aspects of the
11 invention should not be construed to limit the present invention to any particular logic
12 flow or logic implementation. For example, the alternative embodiment of the master to
13 node handshake as depicted in Fig. 6A, shows that described logic may be combined or
14 partitioned into different logic blocks, (e.g., programs, modules, functions, or
15 subroutines) without changing the overall results or otherwise departing from the true
16 scope of the invention. The present invention may thus be embodied in many different
17 forms, including, but not limited to, computer program logic for use with any kind of
18 processor, programmable logic for use with any kind of programmable logic device,
19 discrete components, integrated circuitry including application specific integrated circuits
20 (ASICs), or any other means including any combination thereof. Computer program
21 logic implementing all or part of the functionality described herein may be embodied in
22 various forms, including, but not limited to, source code form, computer executable form,
23 and various intermediate forms (e.g. forms generated by an assembler, compiler, linker,

1 or locator.) Source code may include a series of computer program instructions
2 implemented in any of various programming languages for use with various operating
3 systems or operating environments. The source code may define and use various data
4 structures and communication messages. The source code may be in computer
5 executable form, or it may be in a form convertible into computer executable form. The
6 computer program may be fixed in any form either permanently or transitorily in a
7 tangible storage medium, such as a semiconductor memory device, a magnetic memory
8 device, an optical memory device, a PC card, or other memory device. The computer
9 program may be fixed in any form in a signal that is transmittable to a computer using
10 any of various communication technologies including, but not limited to, analog, digital,
11 optical, wireless, networking, and internetworking technologies. The computer program
12 may be distributed in any form as a removable storage medium with accompanying
13 printed or electronic documentation, preloaded with a computer system (e.g. on system
14 ROM or fixed disk), or distributed from a server or electronic bulletin board over the
15 communication system (e.g., the Internet or World Wide Web).

16
17 The present invention may be used in any network environment where a
18 distributed directory database is needed or utilized, such network being used for any
19 purpose including, but not limited to, computer data storage. Furthermore, although
20 embodiments of the present invention include C++ and JAVA object oriented software,
21 other software could be utilized. Therefore, the scope of the invention is indicated by the
22 appended claims rather than by the foregoing description, and all changes which come

- 1 within the meaning and range of equivalency of the claims are intended to be embraced
- 2 therein.
- 3